

```

UVPKM <-
function(model,fixed.params,which.quantile=0.95,CV.params=NULL,censored.params=NULL,samples=1
000)
{
  sample.vec <- rep(NA,samples)
  MC.matrix <- matrix(NA,nrow=samples,ncol=(length(CV.params)+length(censored.params)))
  colnames(MC.matrix) <- c(names(CV.params),names(censored.params))
  for (this.param in names(CV.params))
  {
    if (!(this.param %in% names(fixed.params))) stop(paste("Cannot find CV.params
parameter",this.param,"in parameter list."))
    if (these.params[[this.param]]>0) MC.matrix[,this.param] <-
rtnorm(samples,mean=fixed.params[[this.param]],sd=fixed.params[[this.param]]*CV.params[[this.param]],
lower=0)
    else
    {
      MC.matrix[,this.param] <- 0
      warning(paste(this.param,"has mean of zero, yielding SD of zero for fixed CV. Parameter value fixed
at zero."))
    }
  }
  for (this.param in names(censored.params))
  {
    if (!(this.param %in% names(fixed.params))) stop(paste("Cannot find censored.params
parameter",this.param,"in parameter list."))
    MC.matrix[,this.param] <-
r.left.censored.norm(samples,mean=fixed.params[[this.param]],sd=fixed.params[[this.param]]*censored.p
arams[[this.param]]$sd,lo=d=censored.params[[this.param]]$lo)
  }
  these.params <- fixed.params
  for (this.sample in 1:samples)
  {
    these.params[colnames(MC.matrix)] <- MC.matrix[this.sample,]
    sample.vec[this.sample] <- eval(call(model,these.params))
  }
  return(quantile(sample.vec,which.quantile))
}

```